Crypto-World

Informační sešit GCUCMP

ISSN 1801-2140

Ročník 15, číslo 9-10/2013 **28. říjen**

9-10/2013

Připravil: Mgr. Pavel Vondruška

Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
http://crypto-world.info/
(1338 registrovaných odběratelů)

Obsah :

	str.
A. Sovietska šifra VIC (J.Kollár)	**2 – 16**
B. Prolamování hash otisků (R.Kümmel)	**17 – 24**
C. Upoutávka na knihu K.Burdy – Aplikovaná kryptografie	**25**
D. Soutěž v luštění / Dosud nevyluštěný dopis českého pobělohorského emigranta Karla Rabenhaupta ze Suché z doby třicetileté války (J.Mírka, P.Vondruška)	**26 – 27**
E. O čem jsme psali za posledních 12 měsíců	**28 – 29**
F. Závěrečné informace	**29**

Příloha: ukázka z knihy Aplikovaná kryptografie
http://crypto-world.info/casop15/Burda_akryptografie.pdf

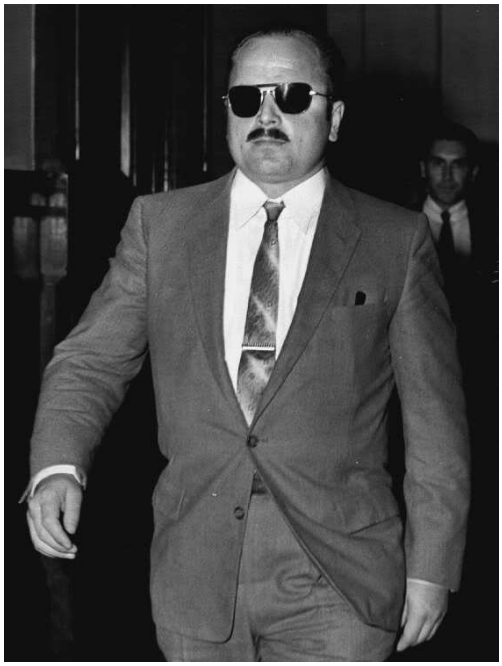
A. Sovietska šifra VIC

Jozef Kollár, jmkollar@math.sk
KMaDG, SvF STU v Bratislave

1 Úvod

V minulom čísle Crypto-World sme si priblížili príbeh sovietskeho agenta Reina Häyhänena. Jedným z dôvodov prečo je Häyhänen zaujímavý aj pre kryptológov je šifra VIC. Bola to sovietska šifra z čias studenej vojny, ktorú tento agent používal. Jednalo sa o šifru typu STT, pričom S bola jedno- dvojmiestna zámena[1] a dve T predstavujú dve transpozície. Prvá z nich bola bežná tabuľková transpozícia a druhá bola špeciálna obrazcová transpozícia podobná československej šifre Zubatka z čias 2. svetovej vojny.

Reino Häyhänen, alias Eugene Maki, alias Victor alebo Vic pôsobil ako agent v USA v rokoch 1952 až 1957.



Obr. 1: Reino Häyhänen (* 14. 5. 1920) počas procesu s Abelom [9]

V pondelok 22. júna 1953 našiel 14-ročný chlapec Jimmy Bozart, predávajúci noviny na Foster Avenue v Brooklyne, preparovanú 5-centovú mincu (*nickel*), v ktorej bol schovaný mikrofilm so zašifrovanou správou. Mincu odovzdal polícii a tá ju posunula ďalej FBI. Táto sa neúspešne pokúšala správu rozlúštiť. To sa im nepodarilo až do mája 1957, keď Häyhänen, potom čo prebehol na americkú stranu, prezradil postup šifrovania a dešifrovania a použité heslá.

[1] V anglickej terminológii *checkerboard cipher*.

V nasledovnom texte podrobne popíšeme šifru VIC, pričom tento popis je kompletne prevzatý z Kahnovho článku [2]. Uvedený článok je pravdepodobne najpresnejší a najpodrobnejší popis šifry VIC. Takisto všetky uvádzané texty a použité heslá sú prevzaté z uvedeného článku a jedná sa o autentické texty a heslá tak, ako boli na mikrofilme v minci. Na webe existujú mnohé ďalšie popisy šifry VIC, ale tie sú zväčša nekompletné, prípadne obsahujú chyby. V Kahnovej knihe *The Codebreakers* je šifra VIC spomenutá a stručne, ale nekompletne, popísaná. A aby situácia okolo tejto šifry bola ešte zmätenejšia, existuje iná šifra VIC[2], ktorú používala východonemecká StaSi a ktorá, okrem mena, nemá s Häyhänenovou šifrou VIC nič spoločné.

2 Postup šifrovania

Pri šifrovaní správy pomocou šifry VIC používal agent heslo, ktoré pozostávalo z piatich častí. Prvé štyri časti boli pevne dané a zrejme[3] boli pridelované agentovi na dlhšiu dobu. Posledná piata časť bolo 5-ciferné číslo, ktoré sa volilo náhodne a malo byť pre každú depešu iné. V Häyhänenovom prípade boli pevné časti hesla nasledovné:

1. Dátum sovietskeho víťazstva nad Japonskom: **3. september 1945**
2. Slovné heslo: **СНЕГОПАД**[4]
3. Slovná fráza: **Только слышно на улице одинокая бродит гармонь**[5]
4. Osobné číslo agenta: **13**

V prípade odchytenej depeše z mince bola piata časť hesla – inicializačný reťazec pozostávajúci z piatich cifier: **20818**.

Z uvedených častí hesla sa potom veľmi komplexným spôsobom vytvárali súradnice pre substitučnú tabuľku, permutácia stĺpcov prvej transpozičnej tabuľky a permutácia stĺpcov druhej transpozičnej tabuľky. Súradnice pre substitučnú tabuľku vždy pozostávali z nejakej permutácie cifier 0 až 9. Dĺžky transpozičných permutácií boli premenlivé a záviseli od toho ako sa zvolila piata, premenlivá časť hesla. Vytváranie substitučnej aj transpozičných permutácii si podrobne popíšeme v samostatnej časti, vzhľadom na jeho zložitosť. Na tomto mieste uvádzame len finálne permutácie použité v Häyhänenovej depeši z mince:

- Súradnice pre substitučnú tabuľku: `5 0 7 3 8 9 4 6 1 2`
- Permutácie stĺpcov prvej transpozičnej tabuľky:
  `14 8 16 2 3 1 13 4 9 10 5 11 15 17 6 12 7`
- Permutácie stĺpcov druhej transpozičnej tabuľky:
  `5 13 2 9 7 6 14 8 3 12 10 11 1 4`

Otvorený text depeše z mince bol nasledovný[6]:

[2] Podľa pána Drobicka skúmajúceho archívy StaSi.
[3] Vyplýva to z popisu v článku [2].
[4] Ruské slovo **Снегопад** znamená *sneženie*.
[5] Jedná sa o prvú slohu (vtedy) v Rusku populárnej piesne *Osamelá harmonika*.
[6] Preklad tohto textu je v prílohe na strane 12.

1. Поздравляем с благополучным прибытием. Потверждаем получение вашего письма в адрес “В” и прочтение письма №1.

2. Для организации прикрытия мы дали указание передать вам три тысячи местных. Перед тем как их вложить в какое лубо дело посоветуйтесь с нами, сообщив характеристику этого дела.

3. По вашей просьбе рецептуру изготовления мягкой пленки и новостей передадим отдельно в месте с письмом матери.

4. Гаммы высылать вам рано. Короткие письма шифруйте, а побольше – [7] делайте со встабками. Все данные о себе, место работы, адрес и т.д. в одной шифровке передать нельзя. Вставки передавайте отдельно.

5. Посылку жене передали лично. С семьей все благополучно. Желаем успеха. Привет от товарищей.

№1/[8]03 Декабря

Teraz si ukážeme postup ako sa tento text šifroval. Popis bude rozdelený na jednotlivé časti, t.j. najskôr substitúcia a potom prvá a následne druhá transpozícia.

3 Substitúcia

Použitá substitúcia sa v slovenskej aj českej terminológii nazýva *jedno- dvojmiestna zámena*. Je to číselná monoalfabetická substitúcia, v ktorej sa znaky otvoreného textu zamieňajú jedno a dvojcifernými číslami. Samotný fakt, že v zašifrovanom texte máme znaky rôznej dĺžky (jedno- a dvojciferné) nijako zvlášť nekomplikuje lúštenie tejto šifry. Lúštenie jedno- dvojmiestnej zámeny je zhruba rovnako zložité ako lúštenie jednoduchej zámeny, v ktorej majú všetky znaky zašifrovaného textu rovnakú dĺžku.

Najskôr si musíme zostaviť substitučnú tabuľku. Táto bude mať 10 stĺpcov a 4 riadky. Stĺpce tabuľky budú očíslované substitučnou permutáciou, čiže v prípade depeše z mince je to: `5 0 7 3 8 9 4 6 1 2`. Prvý riadok tabuľky nebude očíslovaný[9] a súradnice druhého až štvrtého riadku tabuľky budú posledné tri cifry substitučnej permutácie, t.j. v tomto prípade: `6 1 2`.

Do prvého riadku tabuľky sa zľava zapísalo prvých 7 znakov slovného hesla. V prípade depeše z mince je to: **СНЕГОПА**. Posledné tri stĺpce prvého riadku zostávajú voľné. Vo zvyšných troch riadkoch sú potom zapísané ostatné písmena ruskej abecedy okrem znakov s diakritikou a tvrdého znaku (t.j. okrem **Ё, Й, Ъ**) a 7 špeciálnych znakov: **. , П/Л № Н/Ц НТ ПВТ**. Ostatné znaky ruskej abecedy sa do posledných 3 riadkov tabuľky zapisovali po stĺpcoch v poradí podľa abecedy tak, že tretí a piaty stĺpec[10] sa vynechávali. Do tretieho a piateho stĺpca tabuľky sa zapisovalo prvých 6 zo 7 špeciálnych znakov v poradí, v akom sú uvedené vyššie. Posledný špeciálny znak

[7] Na tomto mieste je v depeši slovo **тире**, čo po rusky znamená *pomlčka*.

[8] Namiesto znaku / je v depeši slovo **дробь**, ktoré znamená (aj) *lomítko*.

[9] Nebude mať súradnicu.

[10] Na tomto mieste je v článku [2] zrejme chybná informácia. Ak by bola substitučná tabuľka zostavovaná týmto spôsobom, výrazne by sa tým uľahčilo lúštenie šifry VIC. Ešte sa k tomuto vrátime v záverečných poznámkach na strane 11.

sa zapísal na koniec tabuľky. Takže výsledná substitučná tabuľka pre depešu z mince bola:

	5	0	7	3	8	9	4	6	1	2
—	**С**	**Н**	**Е**	**Г**	**О**	**П**	**А**			
6	Б	Ж	.	К	№	Р	Ф	Ч	Ы	Ю
1	В	З	,	Л	Н/Ц	Т	Х	Ш	Ь	Я
2	Д	И	П/Л	М	НТ	У	Ц	Щ	Э	ПВТ

Pokiaľ sa jedná o zmysel siedmych špeciálnych znakov, tak interpunkčné znamienka sú zrejmé. Zmysel znaku **П/Л** nie je známy, ale zrejme išlo o nejaký prepínač módov. V depeši z mince tento znak nie je použitý. Znak **№** je skratka pre číslo (č.). Znak **Н/Ц** je prepínač módov text/cifry. Zrejme je to skratka ruských slov Нормально/Цифра. Znak **НТ** označuje začiatok depeše a zrejme sa jedná o skratku ruských slov Начало Текста. Znak **ПВТ** znamená *opakovať* a je to skratka ruského slova Повторять.

Čísla, resp. cifry sa pri použitej substitúcii kódovali tak, že pomocou znaku **Н/Ц** sa prepol mód na cifry, potom sa príslušná cifra napísala trikrát po sebe a opäť sa znakom **Н/Ц** prepol mód na text. Takže napr. cifru 1 by sme zapísali ako: `18 111 18`, cifru 7 ako `18 777 18` atď. V depeši z mince sa nachádza viacero cifier. V jednom prípade je omylom cifra 0 zapísaná ako písmeno O. V podrobnom prepise depeše to je vyznačené.

Napokon, ešte predtým ako si uvedieme substituovaný text depeše, treba uviesť, že v šifre VIC sa text depeše pred substitúciou náhodne rozdelil na dve časti. Najskôr sa substituovala druhá časť depeše, potom sa uviedol znak **НТ**, označujúci začiatok textu a následne sa substituovala prvá časť depeše. Toto opatrenie malo sťažiť lúštenie tým, že na začiatku substituovanej depeše sa nenachádzali obvyklé frázy, avšak dosiahol sa pravý opak a lúštenie depeše sa týmto opatrením zjednodušilo.

Po substituovaní textu depeše za čísla sa ešte počet cifier doplnil na najbližší vyšší násobok 5 nulami (náhodnými ciframi). Depeša z mince má na konci tri nuly (2 1 4).

Depeša z mince, po substituovaní znakov za jedno- a dvojciferné čísla vyzerala nasledovne:

```
9692063696119201223612541320296341040207976972541911154231969201961512
6620237519061146797697251972363463201415138602019111563463871320658257
1389858157192920197511504232017588652620151446946319769205192063292119
8382571346718333186798154167209698511657697247919296929201038198151370
2012231236382091370632020081585197209769725425202381925713110815237519
7592051123823234197692067184441867342323611561561134191115423694086763
8698196320792051123416206469292019717498658131116719206972571342019758
1551941563423206715572540061785765717237519869465819611742569752019672
5671582508201620646981563797697254154191107131110126715519415632097697
2541542019781925713110867185551867985611363296070797697254132013206608
6755723117201557651343898132966086760713472329597144679692015719819198
1546920267206818111182569865118183331825763465691228181111867981025694
1513127235651343898132966061239692065611920723679825191576960254723981
3296670207154167389205112341542569751717152215171720969866197020792051
1234681811118671822218672513128693402010424202021 4
```

Počet znakov (aj s nulami na konci) je 1030 a je to zapísané veľmi neprehľadne. Podrobne a prehľadne rozpísaná substitúcia depeše je uvedená v prílohe na strane 13.

9	6	0	3	3	1	8	3	6	6	4	6	9	0	4	7	5
14	**8**	**16**	**2**	**3**	**1**	**13**	**4**	**9**	**10**	**5**	**11**	**15**	**17**	**6**	**12**	**7**
9	6	9	2	0	6	3	6	9	6	1	1	9	2	0	1	2
2	3	6	1	2	5	4	1	3	2	0	2	9	6	3	4	1
0	4	0	2	0	7	9	7	6	9	7	2	5	4	1	9	1
1	1	5	4	2	3	1	9	6	9	2	0	1	9	6	1	5
1	2	6	6	2	0	2	3	7	5	1	9	0	6	1	1	4
6	7	9	7	6	9	7	2	5	1	9	7	2	3	6	3	4
6	3	2	0	1	4	1	5	1	3	8	6	0	2	0	1	9
1	1	1	5	6	3	4	6	3	8	7	1	3	2	0	6	5
8	2	5	7	1	3	8	9	8	5	8	1	5	7	1	9	2
9	2	0	1	9	7	5	1	1	5	0	4	2	3	2	0	1
7	5	8	8	6	5	2	6	2	0	1	5	1	4	4	6	9
4	6	3	1	9	7	6	9	2	0	5	1	9	2	0	6	3
2	9	2	1	1	9	8	3	8	2	5	7	1	3	4	6	7
1	8	3	3	3	1	8	6	7	9	8	1	5	4	1	6	7
2	0	9	6	9	8	5	1	1	6	5	7	6	9	7	2	4
7	9	1	9	2	9	6	9	2	9	2	0	1	0	3	8	1
9	8	1	5	1	3	7	0	2	0	1	2	2	3	1	2	3
6	3	8	2	0	9	1	3	7	0	6	3	2	0	2	0	0
8	1	5	8	5	1	9	7	2	0	9	7	6	9	7	2	5
4	2	5	2	0	2	3	8	1	9	2	5	7	1	3	1	1
0	8	1	5	2	3	7	5	1	9	7	5	9	2	0	5	1
1	2	3	8	2	3	2	3	4	1	9	7	6	9	2	0	6
7	1	8	4	4	4	1	8	6	7	3	4	2	3	2	3	6
1	1	5	6	1	5	6	1	1	3	4	1	9	1	1	1	5
4	2	3	6	9	4	0	8	6	7	6	3	8	6	9	8	1
9	6	3	2	0	7	9	2	0	5	1	1	2	3	4	1	6
2	0	6	4	6	9	2	9	2	0	1	9	7	1	7	4	9
8	6	5	8	1	3	1	1	1	6	7	1	9	2	0	6	9
7	2	5	7	1	3	4	2	0	1	9	7	5	8	1	5	5
1	9	4	1	5	6	3	4	2	3	2	0	6	7	1	5	5
7	2	5	4	0	0	6	1	7	8	5	7	6	5	7	1	7
2	3	7	5	1	9	8	6	9	4	6	5	8	1	9	6	1
1	7	4	2	5	6	9	7	5	2	0	1	9	6	7	2	5
6	7	1	5	8	2	5	0	8	2	0	1	6	2	0	6	4
6	9	8	1	5	6	3	7	9	7	6	9	7	2	5	4	1
5	4	1	9	1	1	0	7	1	3	1	1	1	0	1	2	6
7	1	5	5	1	9	4	1	5	6	3	2	0	9	7	6	9
7	2	5	4	1	5	4	2	0	1	9	7	8	1	9	2	5
7	1	3	1	1	0	8	6	7	1	8	5	5	5	1	8	6
7	9	8	5	6	1	1	3	6	3	2	9	6	0	7	0	7
9	7	6	9	7	2	5	4	1	3	2	0	1	3	2	0	6
6	0	8	6	7	5	5	7	2	3	1	1	7	2	0	1	5
5	7	6	5	1	3	4	3	8	9	8	1	3	2	9	6	6
0	8	6	7	6	0	7	1	3	4	7	2	3	2	9	5	9
7	1	4	4	6	7	9	6	9	2	0	1	5	7	1	9	8
1	9	1	9	8	1	5	4	6	9	2	0	2	6	7	2	0
6	8	1	8	1	1	1	1	8	2	5	6	9	8	6	5	1
1	8	1	8	3	3	3	1	8	2	5	7	6	3	4	6	5
6	9	1	2	2	8	1	8	1	1	1	1	8	6	7	9	8
1	0	2	5	6	9	4	1	5	1	3	1	2	7	2	3	5
6	5	1	3	4	3	8	9	8	1	3	2	9	6	6	0	6
1	2	3	9	6	9	2	0	6	5	6	1	1	9	2	0	7
2	3	6	7	9	8	2	5	1	9	1	5	7	6	9	6	0
2	5	4	7	2	3	9	8	1	3	2	9	6	6	7	0	2
0	7	1	5	4	1	6	7	3	8	9	2	0	5	1	1	2
3	4	1	5	4	2	5	6	9	7	5	1	7	1	7	1	5
2	2	1	5	1	7	1	7	2	0	9	6	9	8	6	6	1
9	7	0	2	0	7	9	2	0	5	1	1	2	3	4	6	8
1	8	1	1	1	1	8	6	7	1	8	2	2	2	1	8	6
7	2	5	1	3	1	2	8	6	9	3	4	0	2	0	1	0
4	2	4	2	0	2	0	2	1	4							

Tabuľka 1: Prvá transpozičná tabuľka pre depešu z mince

4 Prvá transpozícia

Prvá transpozícia bolo obyčajná tabuľková transpozícia s obdĺžnikovou tabuľkou. Šírka tabuľky bola daná permutáciou stĺpcov. Substituovaná depeša sa do transpozičnej tabuľky zapisovala po riadkoch zľava doprava, zhora nadol a čítala sa po stĺpcoch zhora nadol. Poradie čítania stĺpcov určovala permutácia. To ako vypadala prvá transpozičná tabuľka pre depešu z mince možno vidieť na strane 6. Použitá permutácia stĺpcov tabuľky je zapísaná tučným písmom v druhom riadku zhora. V riadku nad ňou je zapísaná číselná postupnosť, ktorej vyčíslením je táto permutácia. Tvorbu hesla si popíšeme ďalej v samostatnej časti. Začiak a koniec depeše po prvej transpozícii potom majú tvar:

6 5 7 3 0 9 4 3 3 7.....6 9 6 6 5 1 8 3 2 2

Takto zašifrovaná depeša sa ďalej šifrovala pomocou druhej transpozície.

5 Druhá transpozícia

Druhá transpozícia bola tiež tabuľková transpozícia s obdĺžnikovou tabuľkou, avšak obrazcová. Znamená to, že správa sa do tabuľky zapisovala v určitých obrazcoch. Šírku transpozičnej tabuľky určovala opäť permutácia stĺpcov, t.j. transpozičné heslo. Výstup z prvej transpozície sa do tabuľky druhej transpozície zapisoval po riadkoch zľava doprava a zhora nadol, ale nie po celej šírke riadkov. Druhú transpozičnú tabuľku pre depešu z mince môžme vidieť na strane 9. V tejto tabuľke sú riadky rozdelené na biele a šedé časti, pričom šedé časti tvoria trojuholníky. Prvý z týchto trojuholníkov sa začína v prvom riadku v stĺpci s číslom 1[11]. Potom v každom ďalšom riadku posunieme začiatok šedej časti o jeden stĺpec doprava, až kým neprídeme na koniec riadku. Za tým nasleduje jeden biely riadok plnej šírky a potom sa v ďalšom riadku začína druhý šedý trojuholník v stĺpci s číslom 2. Opäť v nasledujúcich riadkoch posúvame začiatok šedej časti o jeden stĺpec doprava až po koniec riadku, potom spravíme jeden biely riadok plnej šírky atď. Počet riadkov tabuľky je daný šírkou tabuľky a dĺžkou šifrovanej správy.

Výstup z prvej transpozičnej tabuľky sa potom do druhej tabuľky zapisoval tak, že najskôr sa vypĺňala biela časť riadkov tabuľky. Až keď boli všetky biele časti riadkov vyplnené, tak sa zvyšná časť správy zapisovala do šedých častí riadkov, opäť začínajúc prvým riadkom zhora a píšuc v smere zľava doprava.

Transponovaná správa sa potom z druhej tabuľky čítala rovnakým spôsobom ako pri obyčajnej tabuľkovej transpozícii, t.j. po stĺpcoch zhora nadol a poradie čítania stĺpcov je určené permutačným heslom (permutáciou stĺpcov). Pri čítaní správy z druhej transpozičnej tabuľky sa už nerozlišovali biele a šedé časti riadkov. Kompletne zašifrovaná depeša z mince mala potom podobu:

```
14546 36056 64211 08919 18710 71187 71215 02906 66036 10922 11375 61238
65634 39175 37378 31013 22596 19291 17463 23551 88527 10130 01767 12366
16669 97846 76559 50062 91171 72332 19262 69849 90251 11576 46121 24666
05902 19229 56150 23521 51911 78912 32939 31966 12096 12060 89748 25362
43167 99841 76271 31154 26838 77221 58343 61164 14349 01241 26269 71578
31734 27562 51236 12982 18089 66218 22577 09454 81216 71953 26986 89779
54197 11990 23881 48884 22165 62992 36449 41742 30267 77614 31565 30902
85812 16112 93312 71220 60369 12872 12458 19081 97117 70107 06391 71114
```

[11] Číslo stĺpca určuje permutačné heslo.

```
19459 59586 80317 07522 76509 11111 36990 32666 04411 51532 91184 23162
82011 19185 56110 28876 76718 03563 28222 31674 39023 07623 93513 97175
29816 95761 69483 32951 97686 34992 61109 95090 24092 71008 90061 14790
15154 14655 29011 57206 77195 01256 69250 62901 39179 71229 23299 84164
45900 42227 65853 17591 60182 06315 65812 01378 14566 87719 92507 79517
99651 82155 58118 67197 30015 70687 36201 56531 56721 26306 87185 91796
51341 07796 76655 62716 33588 21932 16224 87721 85519 23191 20665 45140
66098 60959 71521 02334 21212 51110 85227 98768 11125 05321 53152 14191
12166 12715 03116 43041 74822 72759 29130 21947 15764 96851 22370 11391
83520 62297
```

Ak si pozorne porovnáte zašifrovanú depešu s fotografiu depeše z mince na strane 13, tak zistíte, že jediný rozdiel je v piatej pätici od konca depeše. Do skutočnej depeše sa na určenú pozíciu od konca depeše (v tomto prípade na piatu) zapisovala piata[12], premenlivá časť hesla. Bol to 5-ciferný inicializačný reťazec, ktorý mal v popisovanej depeši tvar: **20818**. Preto je depeša z mince o jednu päticu dlhšia ako horeuvedená zašifrovaná depeša. Inicializačný reťazec sa mal voliť pre každú depešu náhodne a podľa možnosti iný ako pre iné depeše.

6 Tvorba permutácii z hesla

Ako sme už uviedli v predošlom texte, heslo šifry VIC pozostávalo zo štyroch pevných a nemenných častí a jednej náhodne volenej časti – inicializačného reťazca. Jednotlivé časti hesla pre depešu z mince sú uvedené na strane 3. Teraz si popíšeme ako sa z týchto častí hesla zostavovali permutácie pre substitučnú a transpozičné tabuľky. V tejto časti sú nielen popis, ale aj označenia prevzaté z Kahnovho článku [2].

Tvorba hesla pri šifre VIC sa začínala dátumom, v našom prípade 3. septembrom 1945. Dátum sa zapísal v číselnej podobe, v tvare `dDmMYYYY`, čiže dostávame 391945. Dĺžka dátumu podľa uvedeného formátu môže byť 6 až 8 cifier.

Posledná cifra dátumu indikuje, na ktorej pozícii od konca depeše bude umiestnená pätica s inicializačným reťazcom. V našom prípade je to 5, čiže piata pätica od konca depeše bude inicializačný reťazec.

Ďalej si zoberieme inicializačný reťazec 20818 (ozn. ako riadok A) a odčítame od neho prvých 5 cifier dátumu 39194 (riadok B). Odčítanie sa robí modulo 10, t.j. bez prenosu desiatky:

```
  A:   2 0 8 1 8
−B:   3 9 1 9 4
------------------
  C:   9 1 7 2 4
```

Výsledné 5-ciferne číslo teraz reťazovým sčítaním modulo 10 natiahneme na 10 cifier. Pod reťazovým sčítaním sa myslí to, že 6. cifru dostaneme ako súčet 1. a 2. cifry, 7. cifru dostaneme ako súčet 2. a 3. cifry atď. Takto dostaneme 10-ciferné číslo:

```
9 1 7 2 4 0 8 9 6 4
```

V ďalších krokoch budeme „vyčíslovať“ textové aj číselné reťazce. To znamená, že budeme určovať poradie znakov, alebo cifier v týchto reťazcoch podľa abecedy, prípadne

[12] Zhoda v čísle 5 je v tomto prípade čisto náhodná.

3	0	2	7	4	3	0	4	2	8	7	7	1	2
5	**13**	**2**	**9**	**7**	**6**	**14**	**8**	**3**	**12**	**10**	**11**	**1**	**4**
6	5	7	3	0	9	4	3	3	7	5	7	1	1
9	1	8	9	3	9	1	2	3	3	4	5	4	2
7	9	3	3	6	0	9	6	2	6	1	9	5	0
1	2	1	5	9	2	1	6	1	2	4	1	4	9
5	3	0	1	1	3	1	6	9	0	6	6	6	6
7	1	1	3	2	8	2	0	2	1	5	0	3	1
8	9	3	9	8	8	1	4	6	5	5	1	6	2
3	1	2	7	7	1	6	4	2	6	2	8	0	0
1	2	2	1	2	4	6	1	6	5	9	2	5	6
7	0	5	7	1	8	1	1	9	3	0	0	6	0
3	6	9	5	2	8	2	5	8	1	1	6	6	8
4	6	6	2	4	8	7	1	4	5	1	3	4	9
2	5	1	9	5	4	1	5	9	6	5	1	2	7
7	4	9	8	8	2	5	3	9	7	7	5	1	4
5	5	2	1	1	2	0	2	0	2	2	6	1	8
6	1	9	6	9	1	3	9	2	1	0	5	0	2
2	4	1	9	0	6	1	1	5	2	6	8	8	5
5	0	1	5	8	5	1	1	1	6	7	1	9	3
1	6	7	7	1	6	6	8	1	3	7	2	1	6
2	6	4	6	9	2	4	4	1	0	1	0	9	2
3	0	6	1	7	9	3	2	5	6	9	1	1	4
6	9	3	6	1	9	0	3	7	8	5	3	8	3
1	8	2	9	1	2	4	1	6	7	0	7	7	1
2	6	3	4	7	3	1	6	4	1	1	8	1	6
9	0	5	8	7	6	7	2	6	8	2	1	0	7
8	9	5	3	0	4	4	8	1	5	5	4	7	9
2	5	1	3	1	4	8	2	2	9	6	5	1	9
1	9	8	2	0	9	2	0	1	1	6	6	1	8
8	7	8	9	7	4	2	1	2	7	9	6	8	4
0	1	5	5	0	1	7	1	4	9	2	8	7	1
8	5	2	1	6	7	2	1	6	6	5	7	7	7
9	2	7	9	3	4	7	9	6	5	0	7	1	6
6	1	1	7	9	2	5	1	6	1	6	1	2	2
6	0	0	6	1	3	9	8	0	3	2	9	1	7
2	2	1	8	7	0	2	5	5	4	9	9	5	1
1	3	3	6	1	2	9	5	9	1	0	2	0	3
8	3	0	3	1	6	1	6	0	0	1	5	2	1
2	4	0	4	1	7	3	1	2	7	3	0	9	1
2	2	1	9	4	7	0	1	1	7	9	7	0	5
5	1	7	9	1	7	2	0	9	9	1	7	6	4
7	2	6	2	9	6	1	2	2	6	7	9	6	2
7	1	7	6	4	1	9	8	2	7	9	5	6	6
0	2	1	1	5	4	4	8	9	6	7	1	0	8
9	5	2	1	9	3	7	7	5	6	1	7	3	3
4	1	3	0	5	1	1	6	6	5	2	9	6	8
5	1	6	9	9	5	5	7	1	5	2	9	1	7
4	1	6	9	5	6	7	6	5	6	9	6	0	7
8	0	1	5	8	5	6	7	0	2	2	5	9	2
1	8	6	0	6	3	4	1	2	7	3	1	2	2
2	5	6	9	8	0	9	8	3	1	2	8	2	1
1	2	6	0	0	9	6	0	5	6	9	2	1	5
6	2	9	2	3	0	8	3	2	3	9	1	1	8
7	7	9	4	1	2	5	5	1	3	8	5	3	3
1	9	7	0	7	8	1	6	5	5	4	5	7	4
9	8	8	9	0	5	2	3	1	8	1	5	5	3
5	7	4	2	7	8	2	2	9	8	6	8	6	6
3	6	6	7	5	1	3	8	1	2	4	1	1	1
2	8	7	1	2	2	7	2	1	1	4	1	2	1
6	1	6	0	2	1	0	2	7	9	5	8	3	6
9	1	5	0	7	6	1	2	8	3	9	6	8	4
8	1	5	8	6	1	1	3	9	2	0	7	6	1
6	2	9	9	5	1	3	1	1	1	0	1	5	4
8	5	5	0	0	2	9	6	2	6	4	9	6	3
9	0	0	0	9	9	1	7	3	2	2	7	3	4
7	5	0	6	1	3	8	4	2	2	2	3	4	9
7	3	6	1	1	3	3	3	9	4	2	0	3	0
9	2	2	1	1	1	5	9	3	8	7	0	9	1
5	1	9	4	1	2	2	0	9	7	6	1	1	2
4	5	1	7	1	7	0	2	3	7	5	5	7	4
1	3	1	9	3	1	6	3	1	2	8	7	5	1
9	1	7	0	6	2	2	0	9	1	5	0	3	2
7	5	1	1	9	2	2	7	6	8	3	6	7	6
1	2	7	5	9	0	9	6	6	5	1	8	3	2
1	1	2	1	0	6	7	2						

Tabuľka 2: Druhá transpozičná tabuľka pre depešu z mince

podľa hodnoty cifier. Prí číselných reťazcoch si treba dať pozor na to, že nula sa berie ako číslo 10. Takže poradie cifier zoradených podľa hodnôt od najmenšej po najväčšiu je: 1 2 3 4 5 6 7 8 9 0.

V nasledujúcom kroku sa zobralo prvých 20 písmen slovnej frázy[13] (riadok D). Týchto 20 znakov sa rozdelí na dve časti po 10 znakov a znaky každej tejto časti sa vyčíslia podľa poradia písmen v ruskej abecede (riadok E). Pod čísla v ľavej časti napíšme 10-ciferné číslo, ktoré sme dostali reťazovým sčítaním riadku C (str. 8). Pod čísla v pravej časti zapíšeme cifry podľa hodnôt (1…0). Cifry v ľavej časti riadkov E a F sčítame modulo 10 a dostávame riadok G.

D:	Т	О	Л	Ь	К	О	С	Л	Ы	Ш	Н	О	Н	А	У	Л	И	Ц	Е	Г
E:	7	4	2	0	1	5	6	3	9	8	6	8	7	1	9	5	4	0	3	2
+F:	9	1	7	2	4	0	8	9	6	4	1	2	3	4	5	6	7	8	9	0
G:	6	5	9	2	5	5	4	2	5	2										

Každú cifru v riadku G teraz vyhľadáme v pravej časti riadku F a nahradíme ju cifrou, ktorá sa nachádza nad ňou v riadku E. Takto dostaneme riadok H a jeho vyčíslením riadok J:

H:	5	9	3	8	9	9	1	8	9	8
J:	3	7	2	4	8	9	1	5	0	6

Riadok J použijeme až v ďalšom kroku. Teraz 10-ciferné číslo z riadku H reťazovým sčítaním modulo 10 predĺžime na 60 cifier. Novovzniknuté cifry budeme zapisovať v riadkoch po 10. Takto dostávame riadky K až P:

H:	5	9	3	8	9	9	1	8	9	8
K:	4	2	1	7	8	0	9	7	7	2
L:	6	3	8	5	8	9	6	4	9	8
M:	9	1	3	3	7	5	0	3	7	7
N:	0	4	6	0	2	5	3	0	4	7
P:	4	0	6	2	7	8	3	**4**	**1**	1

Šírky prvej a druhej transpozičnej tabuľky určíme tak, že vezmeme posledné dve rôzne cifry v riadku P a pričítame k nim osobné číslo agenta. V horeuvedenej tabuľke sú tieto dve cifry 4 a 1 zvýraznené tučným písmom. Šírky transpozičných tabuliek budú:

$$\text{šírka 1. tabuľky:} \quad 4 + 13 = 17$$
$$\text{šírka 2. tabuľky:} \quad 1 + 13 = 14$$

V článku [2], z ktorého je prebratý tento popis, je uvedené, že sa vezmú cifry na 8. a 9. pozícii v riadku P. Ako alternatívna varianta je uvedené, že sa vezmú posledné dve rôzne cifry (čo je v tomto prípade to isté ako 8. a 9. cifra). Ťažko povedať, ktorá z týchto dvoch možností je správna a zrejme ani sám Häyhänen si na to nevedel pri výsluchoch spomenúť. Ja osobne sa z čisto psychologických dôvodov skôr prikláňam k variante posledných dvoch rôznych cifier. Z hľadiska bezpečnosti šifry je úplne jedno[14], či transpozičné tabuľky majú rovnakú alebo rôznu šírku. Ako autor šifry by som ale radšej zvolil tabuľky rôznej šírky.

[13] Tretia časť hesla na strane 3.

[14] Samozrejme stále hovoríme o šifre VIC.

Takže máme už šírky oboch transpozičných tabuliek a teraz si musíme zostrojiť transpozičné heslá (permutácie). Tie dostaneme z riadkov K až P tak, že z nich vyberieme potrebný počet cifier (v našom prípade $17 + 14$). Cifry budeme čítať po stĺpcoch zhora nadol a poradie stĺpcov nám určí permutácia z riadku J, ktorá je vyčíslením riadku H. Z takto vybratých cifier bude prvých 17 tvoriť riadok Q a ďalších 14 cifier bude tvoriť riadok R:

```
Q:   9 6 0 3 3 1 8 3 6 6 4 6 9 0 4 7 5
R:       3 0 2 7 4 3 0 4 2 8 7 7 1 2
```

Vyčíslením riadkov Q a R dostávame permutácie stĺpcov pre 1. a 2. transpozičnú tabuľku. Pod vyčíslením riadkov máme na mysli, rovnako ako predtým, určenie poradia cifier na riadkoch. Pritom opäť cifra nula má najvyššiu hodnotu.

```
1. transpozícia:  14 8 16 2 3 1 13 4 9 10 5 11 15 17 6 12 7
2. transpozícia:        5 13 2 9 7 6 14 8 3 12 10 11 1 4
```

Takže máme obe transpozičné heslá (permutácie). Zostáva nám ešte vygenerovať permutáciu pre substitučnú tabuľku. Tú dostaneme ako vyčíslenie cifier na riadku P a zapíšeme ju ako riadok S:

```
P:   4  0  6  2  7  8  3  4  1  1
S:   5  0  7  3  8  9  4  6  1  2
```

Týmto sme kompletne ukončili proces vytvárania permutácií pre substitučnú aj obe transpozičné tabuľky.

7 Záverečné poznámky

7.1 Substitučná tabuľka

Na strane 4 bolo spomenuté, že pri zápise abecedy do substitučnej tabuľky sa vynechávali tretí a piaty stĺpec. Takto to je uvedené v článku [2]. Ak by sa to skutočne robilo takto, t.j. ak by vynechávané stĺpce boli pevne dané, tak by sa tým zjednodušilo lúštenie šifry. Poznali by sme totiž pozície špeciálnych znakov v substitučnej tabuľke a vedeli by sme ich potom identifikovať v zašifrovanej depeši.

Pravdepodobnejšie je, že v Kahnovom článku je chyba. Zdrojom tejto chyby môže byť či už úmyselná, alebo neúmyselná chybná informácia od Häyhänena pri popise šifry VIC. To, ktoré stĺpce sa vynechávali, zrejme nejakým spôsobom záviselo od použitého hesla. V Häyhänenovom prípade to skutočne mohol byť 3. a 5. stĺpec. Čísla týchto stĺpcov mohli byť určené napríklad prvou a poslednou cifrou dátumu (**3**91945), prípadne nejakým zložitejším spôsobom.

Ak by sme pri lúštení nepoznali čísla vynechaných stĺpcov, museli by sme vyskúšať viacero možností. Pri výkone dnešných počítačov by sa tým lúštenie nijako dramaticky neskomplikovalo, ale v časoch, keď šifra VIC bola aktuálna, by to bola významná komplikácia.

7.2 Druhá transpozičná tabuľka

V popise tvorby druhej šifrovacej tabuľky v [2] nie je uvedené čo robiť ak je šifrovaná správa dlhšia ako transpozičná tabuľka s toľkými šedými trojuholníkmi, koľko má stĺpcov. V prípade depeše z mince tento problém nenastal, pretože jej dĺžka je taká, že druhá transpozičná tabuľka má len 9 zo 14 možných šedých trojuholníkov a pritom deviaty šedý trojuholník ani nie je kompletný a kompletne vyplnený. Pre tento problém sa prirodzene ponúkajú dve možné riešenia:

1. Obmedziť dĺžku šifrovanej správy tak, aby sa zmestila do transpozičnej tabuľky s kompletnou sadou šedých trojuholníkov. Znamenalo by to, že dlhšie správy by bolo nutné rozdeľovať a šifrovať samostatne.

2. Po vyčerpaní kompletnej sady šedých trojuholníkov v druhej transpozičnej tabuľke by sa začali tvoriť ďalšie šedé trojuholníky, opäť začínajúc v stĺpci s číslom 1 a ďalej podľa už uvedeného postupu.

7.3 Osobné číslo

V snahe zvýšiť bezpečnosť šifry, zmenili Häyhänenoví nadriadení v roku 1956 jeho osobné číslo z 13 na 20. Tým sa v priemere zväčšili šírky oboch transpozičných tabuliek. Aby toto opatrenie bolo vždy realizovateľné, muselo sa aj reťazové sčítanie čísel z riadku H (str. 10) predĺžiť zo 60 na 70 cifier (o 1 riadok). Toto opatrenie však lúštenie šifry nijako nezmení, len ho môže mierne predĺžiť.

7.4 Lúštenie šifry VIC

V závere článku [2] pán Kahn píše, že pri znalosti šifrovacieho algoritmu a dostatočnom množstve odchytených depeší, je možné tieto analyzovať a šifru VIC lúštiť[15]. Toto tvrdenie by som označil za nie celkom presné. Pri mojom „hraní sa“ so šifrou VIC som totiž zistil, že pri znalosti šifrovacieho algoritmu tak ako je popísaný v Kahnovom, a teda aj v tomto, článku, je šifra VIC pomerne jednoducho lúštitelná už pri jedinej odchytenej depeši. To, že kryptoanalytici FBI neuspeli, bolo spôsobené tým, že nevedeli s čím presne majú dočinenia. Ak by poznali postup šifrovania, tak by Häyhänenovu depešu určite rozlúštili.

Príloha A

Slovenský preklad textu depeše z mince:

1. Gratulujeme vám k úspešnému príchodu. Týmto potvrdzujeme príjem vášho dopisu na adresu „V“ a prečítanie dopisu č. 1.

2. Na zabezpečenie vášho krytia sme dali príkaz doručiť vám 3000 dolárov. Predtým než ich investujete do akéhokoľvek podnikania, informujte nás o charaktere tohto podnikania a poraďte sa s nami.

[15] Toto je približné parafrázovanie podstatne dlhšej poznámky, ale jej zmysel je zachovaný.

3. Vami požadovanú receptúru na výrobu mäkkého filmu a novinky vám doručíme inou cestou, spolu s dopisom od matky.

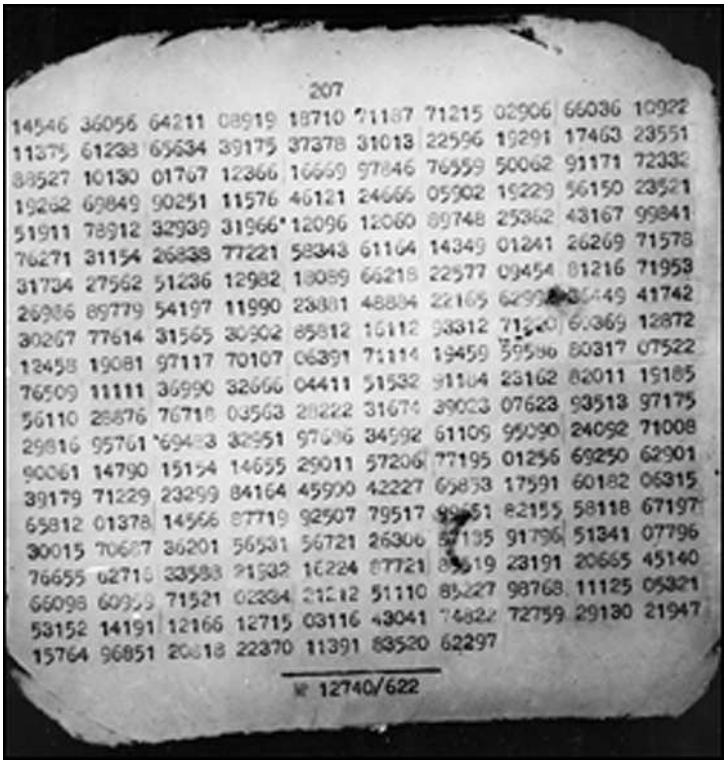
4. Je skoro na to, aby sme vám posielali *gammy*[16]. Kratšie dopisy šifrujte a dlhšie robte po častiach[17]. Údaje o vás, ako sú miesto kde pracujete, adresa atď. nesmiete uvádzať spolu v jednej depeši. Jednotlivé časti[18] doručujte oddelene.

5. Balík sme vašej žene doručili osobne. Vaša rodina je v poriadku. Prajeme vám úspech. Súdruhovia vás pozdravujú.

č. 1 / 3. decembra

Príloha B

Fotografia skutočnej depeše z mince:



Príloha C

Podrobný prepis substitúcie depeše z mince. Farebné označenia v tabuľke sú nasledovné:

- **Zelenou** farbou je označený skutočný začiatok textu (symbol **НТ**) depeše.

[16]Pojmom *gamma* sa v ruskej terminológii označovali OTP tabuľky.

[17]Na tomto mieste si nie som s prekladom celkom istý. Slovo **вставка** znamená v ruštine *vložka* (napr. medzera vložená do textu, obraz vložený do rámu, alebo tuha do pera), ale môže to znamenať aj *inzerát*. Zrejme sa tým myslelo delenie depeší na časti, ale nedá sa vylúčiť ani nejaká steganografická metóda, ako napr. skrývanie dopisov v dutých šróboch, preparovaných baterkách a pod., na čo bol Abel expert.

[18]Opäť je tu použité slovo **вставки**, viď predošlá poznámka pod čiarou.

- **Žltá** farba označuje zakódované cifry.
- **Modrá** farba označuje symboly, ktoré boli vypísané slovom (**тире** = *pomlčka* a **дробь** = *lomítko*.
- **Fialová** farba označuje miesto, kde bola v depeši, zrejme omylom, zapísaná 0 (cifra) ako veľké O.
- **Šedá** farba označuje doplnené nuly na konci depeše.
- **Červená** farba označuje miesto, kde bol v článku [2] preklep (19 namiesto 69). Tento preklep sa vyskytuje len v prepise depeše v uvedenom článku. V skutočnej depeši je to uvedené správne.

П	Р	И	К	Р	Ы	Т	И	Я	М	Ы	Д	А	Л	И	У
9	69	20	63	69	61	19	20	12	23	61	25	4	13	20	29
К	А	З	А	Н	И	Е	П	Е	Р	Е	Д	А	Т	Ь	В
63	4	10	4	0	20	7	9	7	69	7	25	4	19	11	15
А	М	Т	Р	И	Т	Ы	С	Я	Ч	И	М	Е	С	Т	Н
4	23	19	69	20	19	61	5	12	66	20	23	7	5	19	0
Ы	Х	.	П	Е	Р	Е	Д	Т	Е	М	К	А	К	И	Х
61	14	67	9	7	69	7	25	19	7	23	63	4	63	20	14
В	Л	О	Ж	И	Т	Ь	В	К	А	К	О	Е	Л	И	Б
15	13	8	60	20	19	11	15	63	4	63	8	7	13	20	65
О	Д	Е	Л	О	П	О	С	О	В	Е	Т	У	И	Т	Е
8	25	7	13	8	9	8	5	8	15	7	19	29	20	19	7
С	Ь	С	Н	А	М	И	,	С	О	О	Б	Щ	И	В	Х
5	11	5	0	4	23	20	17	5	8	8	65	26	20	15	14
А	Р	А	К	Т	Е	Р	И	С	Т	И	К	У	Э	Т	О
4	69	4	63	19	7	69	20	5	19	20	63	29	21	19	8
Г	О	Д	Е	Л	А	.	Н/Ц	333	Н/Ц	.	П	О	В	А	Ш
3	8	25	7	13	4	67	18	333	18	67	9	8	15	4	16
Е	И	П	Р	О	С	Ь	Б	Е	Р	Е	Ц	Е	П	Т	У
7	20	9	69	8	5	11	65	7	69	7	24	7	9	19	29
Р	У	И	З	Г	О	Т	О	В	Л	Е	Н	И	Я	М	Я
69	29	20	10	3	8	19	8	15	13	7	0	20	12	23	12
Г	К	О	И	П	Л	Е	Н	К	И	И	Н	О	В	О	С
3	63	8	20	9	13	7	0	63	20	20	0	8	15	8	5
Т	Е	И	П	Е	Р	Е	Д	А	Д	И	М	О	Т	Д	Е
19	7	20	9	7	69	7	25	4	25	20	23	8	19	25	7
Л	Ь	Н	О	В	М	Е	С	Т	Е	С	П	И	С	Ь	М
13	11	0	8	15	23	7	5	19	7	5	9	20	5	11	23
О	М	М	А	Т	Е	Р	И	.	Н/Ц	444	Н/Ц	.	Г	А	М
8	23	23	4	19	7	69	20	67	18	444	18	67	3	4	23
М	Ы	В	Ы	С	Ы	Л	А	Т	Ь	В	А	М	Р	А	Н
23	61	15	61	5	61	13	4	19	11	15	4	23	69	4	0
О	.	К	О	Р	О	Т	К	И	Е	П	И	С	Ь	М	А
8	67	63	8	69	8	19	63	20	7	9	20	5	11	23	4

Ш	И	Ф	Р	У	И	Т	Е	,	А	П	О	Б	О	Л	Ь
16	20	64	69	29	20	19	7	17	4	9	8	65	8	13	11
Ш	Е	Т	И	Р	Е	Д	Е	Л	А	И	Т	Е	С	О	В
16	7	19	20	69	7	25	7	13	4	20	19	7	5	8	15
С	Т	А	В	К	А	М	И	.	В	С	Е	Д	А	Н	Н
5	19	4	15	63	4	23	20	67	15	5	7	25	4	0	0
Ы	Е	О	С	Е	Б	Е	,	М	Е	С	Т	О	Р	А	Б
61	7	8	5	7	65	7	17	23	7	5	19	8	69	4	65
О	Т	Ы	,	А	Д	Р	Е	С	И	Т	.	Д	.	В	О
8	19	61	17	4	25	69	7	5	20	19	67	25	67	15	8
Д	Н	О	И	Ш	И	Ф	Р	О	В	К	Е	П	Е	Р	Е
25	0	8	20	16	20	64	69	8	15	63	7	9	7	69	7
Д	А	В	А	Т	Ь	Н	Е	Л	Ь	З	Я	.	В	С	Т
25	4	15	4	19	11	0	7	13	11	10	12	67	15	5	19
А	В	К	И	П	Е	Р	Е	Д	А	В	А	И	Т	Е	О
4	15	63	20	9	7	69	7	25	4	15	4	20	19	7	8
Т	Д	Е	Л	Ь	Н	О	.	Н/Ц	555	Н/Ц	.	П	О	С	Ы
19	25	7	13	11	0	8	67	18	555	18	67	9	8	5	61
Л	К	У	Ж	Е	Н	Е	П	Е	Р	Е	Д	А	Л	И	Л
13	63	29	60	7	0	7	9	7	69	7	25	4	13	20	13
И	Ч	Н	О	.	С	С	Е	М	Ь	Е	И	В	С	Е	Б
20	66	0	8	67	5	5	7	23	11	7	20	15	5	7	65
Л	А	Г	О	П	О	Л	У	Ч	Н	О	.	Ж	Е	Л	А
13	4	3	8	9	8	13	29	66	0	8	67	60	7	13	4
Е	М	У	С	П	Е	Х	А	.	П	Р	И	В	Е	Т	О
7	23	29	5	9	7	14	4	67	9	69	20	15	7	19	8
Т	Т	О	В	А	Р	И	Щ	Е	И	№	Н/Ц	111	Н/Ц	Д	Р
19	19	8	15	4	69	20	26	7	20	68	18	111	18	25	69
О	Б	Ь	О	Н/Ц	333	Н/Ц	Д	Е	К	А	Б	Р	Я	НТ	Н/Ц
8	65	11	8	18	333	18	25	7	63	4	65	69	12	28	18
111	Н/Ц	.	П	О	З	Д	Р	А	В	Л	Я	Е	М	С	Б
111	18	67	9	8	10	25	69	4	15	13	12	7	23	5	65
Л	А	Г	О	П	О	Л	У	Ч	Н	Ы	М	П	Р	И	Б
13	4	3	8	9	8	13	29	66	0	61	23	9	69	20	65
Ы	Т	И	Е	М	.	П	О	Д	Т	В	Е	Р	Ж	Д	А
61	19	20	7	23	67	9	8	25	19	15	7	69	60	25	4
Е	М	П	О	Л	У	Ч	Е	Н	И	Е	В	А	Ш	Е	Г
7	23	9	8	13	29	66	7	0	20	7	15	4	16	7	3
О	П	И	С	Ь	М	А	В	А	Д	Р	Е	С	,	,	В
8	9	20	5	11	23	4	15	4	25	69	7	5	17	17	15
ПВТ	В	,	,	И	П	Р	О	Ч	Т	Е	Н	И	Е	П	И
22	15	17	17	20	9	69	8	66	19	7	0	20	7	9	20
С	Ь	М	А	№	Н/Ц	111	Н/Ц	.	Н/Ц	222	Н/Ц	.	Д	Л	Я
5	11	23	4	68	18	111	18	67	18	222	18	67	25	13	12
О	Р	Г	А	Н	И	З	А	Ц	И	И	*nuly*				
8	69	3	4	0	20	10	4	24	20	20	2 1 4				

Literatúra

[1] Kahn David: The Codebreakers (str. 668-671)
Scribner, 1996

[2] Kahn David: Number One from Moscow
CIA Historical Review Program – odtajnené 1993

[3] Kahn Jeffrey: The Case of Colonel Abel
Journal of National Security, Law & Policy, June 2010

[4] Rocafort W. W.: Colonel Abel's Assistant
CIA Studies in Intelligence, Vol. 3, Issue: Fall, 1959 – odtajnené 1994

[5] Wikipedia: Hollow Nickel Case
`http://en.wikipedia.org/wiki/Hollow_Nickel_Case`

[6] Wikipedia: Rudolf Abel
`http://en.wikipedia.org/wiki/Vilyam_Genrikhovich_Fisher`

[7] Wikipedia: Reino Häyhänen
`http://en.wikipedia.org/wiki/Reino_Häyhänen`

[8] Fotografia: depeša z mince
`http://mentalfloss.com/article/32100/how-nickel-and-paperboy-brought-down-cold-war-spy`

[9] Fotografia: Reino Häyhänen
`http://i1010.photobucket.com/albums/af222/shawncamp/BOS6/BOS-1735/img0005A.jpg`

Práce zaoberajúce sa šifrou VIC

Na FEI STU v Bratislave sa šifrou VIC vo svojich bakalárskych prácach zaoberali dvaja študenti:

1. **Šifra VIC a jej softwérová realizácia (2007)** – autorom práce bol Zoltán Mierka, školiteľom Prof. RNDr. Otokar Grošek, PhD. Práca obsahuje popis šífry VIC a následne jeho softwarovú realizáciu. Žiaľ autor nemal k dispozícii článok [2] a vychádzal len z knihy [1] a webových zdrojov, kde popis nie je presný a kompletný. Oproti Häyhänenovej šifre VIC sú rozdiely v tom, že azbuka je zamenená za latinku, z čoho vyplýva zmena substitučnej tabuľky a okrem toho sa znaky do substitučnej tabuľky zapisovali iným (bezpečnejším) spôsobom.

2. **Kryptoanalýza šifry VIC (2009)** – autorom práce bol Radoslav Čagala, školiteľom Ing. Pavol Zajac, PhD. V tejto práci opäť autor nemal k dispozícii článok [2] a popis šifry VIC prebral z predošlej práce Zoltána Mierku. Autor našiel najslabšie miesto šífry VIC a ukázal útok s využitím tejto slabiny, pričom ho aj softwarovo realizoval. Avšak, aj v dôsledku nepresnosti a nekompletnosti popisu, už nevyužil ďalšie slabiny šifry VIC, a preto bol tento útok časovo pomerne náročný (ale v praxi realizovateľný).

B. Ochrání hashování uživatelská hesla?

R.Kümmel, http://soom.cz/ , ccuminn@soom.cz

Přesto, že se již útoky SQL injection neobjevují na přednějších místech v žebříčcích nejčastějších webových zranitelností, stále se jim i letos daří se svým sedmiprocentním výskytem držet zuby nehty čtrnácté příčky v průzkumu, který pravidelně vydává společnost WhiteHat Security (https://www.whitehatsec.com/resource/stats.html).

V praxi tato informace znamená, že zhruba sedmi webům ze sta mohou útočníci vykrást data z databáze, což určitě není nikterak veselá představa. Obzvláště nepříjemné následky může takový útok mít pro uživatele, kteří nedbají základní bezpečnostní poučky a ve všech webových aplikacích si při registraci volí stejná hesla. Jediný úspěšný průnik do databáze některého z webových serverů pak stačí k tomu, aby nepoučitelný uživatel zcela přišel o svou virtuální identitu. Když si totiž uvědomíme, že v odcizené databázi mohly být vedle různých osobních údajů uloženy také e-mailové adresy, loginy a přístupová hesla, tak nám velmi rychle dojde, že pro útočníka není nic snazšího, než stejné heslo vyzkoušet použít právě pro přístup k samotné e-mailové schránce nebo k různým sociálním sítím. Dnes, kdy je elektronická identita pro mnoho lidí jejich druhým životem, na kterém mnohdy pracovali několik let, je zřejmé, jak nepříjemný pro ně podobný únik dat může být. A nemusí jít vždy pouze o únik způsobený útokem SQL injection. Na denním pořádku jsou také krádeže dat ze strany zaměstnanců firem, nebo případy, kdy společnost vyřadí a prodá starší hardware bez důkladného vymazání uložených dat.

O SQL injekci

SQL injection je útok na vstup aplikace, pomocí kterého může útočník ovlivnit SQL dotaz. Aplikace pak místo dat, pro jejichž zobrazení byla navržena, vrátí jiná v databázi uložená data – například obsah tabulky s uživatelskými účty.

Příklad: Dotaz na webový server
http://www.clanky.cz?id=123
by mohl být na straně serveru pomocí php zpracován následovně:

$id = $_GET[“id”];
$sql = “SELECT * FROM clanky WHERE id=$id“;
$result = mysql_query($sql);

Položme si otázku, co by se stalo, pokud by útočník pozměnit odesílaný dotaz takto:
http://www.clanky.cz?id=123 OR 1=1
SQL dotaz by po tomto zásahu vypadal následovně a vracel by všechny záznamy z tabulky clanky:

SELECT * FROM clanky WHERE id=$id OR 1=1

Následně by bylo možné využít klauzule ORDER BY pro zjištění počtu sloupců a UNION ALL SELECT pro připojení a zobrazení obsahu dalších tabulek.

V tuto chvíli vnímavý čtenář možná namítne, že doba, kdy se hesla v databázích ukládala ve formě prostého textu, je dávno pryč. Dnes přeci mají vývojáři k dispozici hned několik bezpečných hashovacích funkcí a jsou dostatečně poučeni o tom, aby tyto funkce při uchovávání hesel používali. Je ale použití hashů k uchování hesel skutečně dostatečně bezpečným řešením, které dokáže zaručit, že po průniku do databáze nebudou hesla uživatelů prozrazena?

Odhlédneme-li od útoků, při kterých by útočník pozměnil zdrojový kód serverových skriptů tak, aby zaznamenával hesla uživatelů už během jejich přihlašování (tedy ve chvíli, kdy hesla

ještě nejsou zahashována), a budeme-li brát v potaz pouze samotný únik dat z databáze, pak není možné jednoduše konstatovat, zda společně s únikem hashů došlo současně i k prozrazení samotných hesel. K tomu, abychom mohli říci, nakolik je hrozba odhalení původního hesla z jejich otisků, které útočník získal, reálná, se budeme muset do tajů hashovacích funkcí ponořit trochu hlouběji.

Hashovací funkce jsou jednosměrné

To znamená, že pokud heslo jednou proměníme kryptografickými metodami na jeho hash, není následně možné z tohoto otisku zpětně vypočítat jeho původní hodnotu – heslo. V praxi to pak vypadá tak, že ve chvíli, kdy se uživatel zaregistruje do nějaké webové služby, je heslo, které si zvolil, zahashováno a do databáze je uložen právě jen hash (neboli otisk) hesla. Když se chce uživatel následně k aplikaci přihlásit a odešle své heslo, aplikace jej stejným algoritmem opět promění na hash a porovná, zda takto získaný otisk souhlasí s hodnotou hashe, kterou má uloženu v databázi. Pokud si otisky odpovídají, je zřejmé, že uživatel zadal stejná hesla a autentizace může být úspěšně dokončena.

Způsoby prolamování hashů

Otázka tedy zní: Jak se může útočník po krádeži dat dostat k původním heslům, když nemá možnost je vypočítat z odcizených hashů zpět? Jde o zásadní otázku, na kterou budu odpovídat v téměř celém zbytku tohoto článku. V rychlosti si řekněme, že útočník může získat heslo odpovídající konkrétnímu hashi velice snadno, a to tak, že bude testovat shodu hodnot samotných otisků. Tedy tak, že útočník vezme heslo, které chce otestovat, vytvoří z něho hash stejným algoritmem, který byl použit v aplikaci (z níž pochází odcizené hashe) a porovná, zda se získaný otisk shoduje s tím odcizeným. Ve chvíli, kdy si oba hashe odpovídají, může útočník zvolat: „Heuréka!“. Popsaný princip prolamování hashů by se dal rozdělit hned na několik variant útoků, které se běžně používají:

Brutte Force (útok hrubou silou)

Při tomto typu útoku zkouší útočník tvořit hesla ze všech variant použitých znaků (písmena, čísla, symboly). Postupně tedy vyzkouší hesla jako: *a, b, c, ..., aa, ab, ac, ..., ba, bb, bc, ..., aaa, aab, aac, ...* atd. Variant textových řetězců ovšem existuje nekonečné množství, a zda tedy bude heslo v reálném časovém horizontu tímto způsobem prolomeno, záleží pouze na jeho délce a použité množině znaků. Je jasné, že heslo tvořené čtyřmi číslicemi se útočníkovi podaří tímto způsobem najít mnohem dříve, než heslo dlouhé 12 znaků obsahující malá a velká písmena, číslice a nějaký ten speciální znak. Přiložená tabulka znázorňuje dobu potřebnou pro otestování všech možných kombinací řetězců o různých délkách a s různou množinou použitých znaků. Vyplývá z ní, že dostatečně dlouhé a různorodé heslo tomuto typu útoku snadno odolá.

V tabulce uvádím hodnoty pro rychlost crackování 1.000 pokusů za sekundu, což je dnes již směšná hodnota. Vývoj v této oblasti běží totiž neuvěřitelným tempem, a současné systémy využívající výpočetního výkonu až osmi grafických karet dokáží při brutte force útocích vyvinout mnohonásobně vyšší rychlost. Pro představu, nejnovější cluster Jeremiho Gosneye, který byl představen na konci minulého roku na konferenci *Passwords^12* v Oslu, dokáže vyvinout rychlost crackování neuvěřitelných 350 miliard pokusů za sekundu v případě algoritmu NTLM nebo 180 miliard pokusů za sekundu v případě algoritmu MD5.
Zdroj:
http://arstechnica.com/security/2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours

Dictionary attack (slovníkový útok)

Dictionary attack se na rozdíl od brutte force útoku nesnaží tvořit textové řetězce (hesla) skládáním různých znakových variací. Namísto toho je útočný nástroj obdařen slovníkem se seznamem konkrétních slov, jejichž shoda bude během prolamování hesla otestována. Tímto seznamem přitom může být seznam běžných jmen a příjmení, slovník českých slov, seznam jmen pohádkových postaviček, nebo například seznam nejpoužívanějších hesel. Rozdíl oproti brutte force útoku je tedy na první pohled viditelný. Pokud se při brutte force útoku testovaly miliony a miliardy kombinací, pak při slovníkovém útoku se jich otestují pouze stovky nebo tisíce. Dictionary attack tak sice není schopen odhalit všechna hesla, ale útočník díky němu může ve velmi krátké době zjistit alespoň ta hesla, jejichž uživatelé si pro svou ochranu zvolili právě běžné slovo. Seznam nejpoužívanějších hesel najdete v přiložené tabulce.

Nejčastěji používaná hesla

Ve světě

password, 123456, 12345678, abc123, qwerty, monkey, letmein, dragon, 111111, baseball, iloveyou, trustno1, 1234567, sunshine, master, 123123, welcome, shadow, ashley, football, jesus, michael, ninja, mustang, password1

Zdroj: Výsledek studie zkoumající záznamy databází uniklých v roce 2012
http://www.theregister.co.uk/2012/12/03/lame_passwords_still_rife/

V česku

Podle studie, kterou vypracovala UP Olomouc používají čeští uživatelé převážně silná hesla s průměrnou délkou 8,71znaků. Mezi slovníkovými hesly převládají křestní jména včetně zdrobnělin a názvy měst.
Zdroj: *http://prvok.upol.cz/index.php/news/96-according-to-the-centre-for-the-prevention-of-risky-virtual-communication-almost-half-of-the-internet-users-use-a-universal-password*

Rainbow tables

U výše uvedených variant se vždy vzal textový řetězec, který představoval testované heslo, ten se zahashoval a výsledný hash se teprve porovnal se zkoumaným otiskem. To vše v cyklu tak dlouho, dokud nedošlo k nalezení správné kombinace. Poněkud zdlouhavé, nemyslíte? Od výše uvedených variant se proto použití rainbow tables liší v tom, že je již dopředu vygenerovaná databáze dvojic heslo-hash. Při hledání hesla, které by odpovídalo konkrétnímu hashi se tak jen nahlédne do této databáze, zda je v ní tento otisk obsažen. Pokud ano, je útočníkovi obratem vráceno odpovídající heslo.

Velikost rainbow tabulek v komprimované podobě (MD5)

md5_alpha-space	#1-9	24 GB
md5_hybrid2(loweralpha#7-7,numeric#1-3)	#0-0	26 GB
md5_loweralpha	#1-10	180 GB
md5_loweralpha-numeric	#1-10	588 GB
md5_loweralpha-numeric-space	#1-8	16 GB
md5_loweralpha-numeric-space	#1-9	109 GB
md5_loweralpha-numeric-symbol32-space	#1-7	34 GB
md5_loweralpha-numeric-symbol32-space	#1-8	425 GB
md5_loweralpha-space	#1-9	35 GB
md5_mixalpha-numeric	#1-9	1009 GB
md5_mixalpha-numeric-all-space	#1-7	86 GB
md5_mixalpha-numeric-all-space	#1-8	1049 GB
md5_mixalpha-numeric-space	#1-7	18 GB
md5_mixalpha-numeric-space	#1-8	207 GB
md5_numeric	#1-14	91 GB

Zdroj: https://www.freerainbowtables.com/en/tables/

Rainbow tables jsou proto poměrně mocným nástrojem v rukou útočníků, kteří se zmocní uživatelské databáze s otisky hesel. Jejich nevýhodou je ale velikost, kterou tyto databáze zabírají, viz. přiložená tabulka.

Kolize

O kolizích se v posledních letech poměrně hlasitě hovořilo. O co ale ve skutečnosti jde a mají kolize i nějaký vliv na prolamování hesel? Pokud si uvědomíme, že délka hashe je například 32 znaků s množinou například 16-ti různých znaků, pak možných hashů, které z nich lze vytvořit je $V_{32(16)} = 16^{32} = 1\ 208\ 925\ 819\ 614\ 629\ 174\ 706\ 176$. To je samozřejmě hodně velké číslo, ale rozhodně to není nekonečně mnoho variant. Naopak různých hesel, pokud nebudou omezena délkou, mohou uživatelé vytvářet skutečně nekonečné množství. Z toho vyplývá, že různá hesla budou muset generovat stejný hash a bude tak docházet ke kolizím. Vzniklo již několik postupů pro vytváření kolizí například pro algoritmus MD5 a tyto metody se neustále zrychlují. Jednou z významných osobností v této oblasti byl například i náš Vlastimil Klíma.

Zůstaneme-li u algoritmu MD5, je nutno jedním dechem dodat, že pro hesla se těchto postupů zatím stále ještě příliš prakticky využít nedá. Jedním z důvodů je skutečnost, že programy pro hledání kolizí u MD5 generují zatím stejné otisky pouze pro dva předem neznámé řetězce a za druhé, data, která byste tímto způsobem získali, by byla binárního charakteru. S tím, že byste tímto způsobem získali jiné náhradní heslo o délce řekněme do padesáti znaků, proto prostě nepočítejte.

Jiná situace je ovšem u hashů, jejich délka je menší a pravděpodobnost vzniku kolizí mnohonásobně vyšší. Příkladem budiž starší hashovací algoritmus MySQL323, se kterým se můžete stále ještě u některých aplikací setkat. Tento algoritmus generuje hashe o délce šestnácti znaků a existují proto rainbow tabulky obsahující všechny možné kombinace. Najít kolizi v podobě řetězce tvořeného ASCII znaky je v tomto případě mnohem jednodušší, než najít originální heslo. V praxi se proto kolizí u tohoto algoritmu používá zcela běžně.

Jak poznat o jaký typ hashe se jedná

Ve chvíli, kdy se útočník dostane k hashi hesla, je pro něj důležité, aby dokázal rozhodnout, jakým algoritmem byl daný otisk vytvořen. Nejvíce mu v tom napoví délka hashe, která je u různých hashovacích algoritmů odlišná. Napovědět ovšem může také zdroj, ze kterého hash pochází. Jiný typ hashe totiž můžeme očekávat v databázi MS SQL a jiný v MySQL, Oracle, nebo v souboru passwd unixových systémů. Záleží samozřejmě na konkrétní implementaci kryptografie v aplikaci. V některých případech (například u MS SQL), bývá zjištění typu hashe hračkou, protože sám hash dokáže napovědět, jakým algoritmem byl vytvořen. Pojďme si jeden takový záznam z DB prozkoumat.

SQL server 2005 & 2008

0×0100993BF2315F36CC441485B35C4D84687DC02C78B0E680411F

0×0100	Konstantní hlavička určující typ algoritmu
993BF231	Sůl
5F36CC441485B35C4D84687DC02C78B0E680411F	Case sensitive hash SHA1

SQL Server 2000:

0×010034767D5C0CFA5FDCA28C4A56085E65E882E71CB0ED2503412FD54D6119FFF04129A1D72E7C3194F7284A7F3A

0×0100	Konstantní hlavička určující typ

algoritmu		
34767D5C	Sůl	
0CFA5FDCA28C4A56085E65E882E71CB0ED250341	Case senstive hash	SHA1
2FD54D6119FFF04129A1D72E7C3194F7284A7F3A	Upper case hash	

Vidíte, že hlavička zde přímo udává použitý hashovací algoritmus a tím útočníkovi velmi usnadňuje práci. Různé hodnoty, jichž může tato hlavička nabývat, jsou uvedeny v přiložené tabulce. O soli, která je v příkladech zmíněna si povíme níže.

Možné varianty obsahu hlavičky u hashí z MS SQL

```
0x0000   MD5
0x0011   Joomla
0x0021   osCommerce, xt:Commerce
0x0100   SHA1
0x0101   nsldap, SHA-1(Base64), Netscape LDAP SHA
0x0111   nsldaps, SSHA-1(Base64), Netscape LDAP SSHA
0x0112   Oracle 11g
0x0121   SMF > v1.1
0x0122   OSX v10.4, v10.5, v10.6
0x0131   MSSQL(2000)
0x0132   MSSQL(2005, 2008)
0x0300   MySQL
0x0400   phpass, MD5(Wordpress), MD5(phpBB3)
0x0500   md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
0x0900   MD4
0x1000   NTLM
0x1100   Domain Cached Credentials, mscash
0x1400   SHA256
0x1500   descrypt, DES(Unix), Traditional DES
0x1600   md5apr1, MD5(APR), Apache MD5
0x1700   SHA512
0x1722   OSX v10.7
0x2100   Domain Cached Credentials2, mscash2
0x2400   Cisco-PIX MD5
0x2600   Double MD5
0x2611   vBulletin < v3.8.5
0x2711   vBulletin > v3.8.5
0x2811   IPB2+, MyBB1.2+
0x3000   LM
0x3100   Oracle 7-10g, DES(Oracle)
```

Příklad: Jak vypadají hashe získané různými algoritmy

LM:	4D5A1DB67431A871AAD3B435B51404EE
NTLM:	4CE828B8064CC2BE60FFE9FAC8DC57ED
MySQL 323:	4E633CF914A735A0
MySQL:	AADB89769FF364D8537700AE065872FDA6E980F3
SHA1:	e017b5464f820a6c1bb5e9f6d711a667a80d8ea
CRC32:	216a42cb
ADLER32:	0641021c
CRC32A:	e08e750f
GOST:	0d1e8f4bc119f3fba4cd10a98181910429d0d926d454e3730a49f128d4348e14
MD2:	7ab20b2bbef9381de37c4d9badf69a33
MD4:	8e3134b595dc9056e0b2ea4250b73ba5
MD5:	955db0b81ef1989b4a4dfeae8061a9a6
SHORT_MD5:	qum9egmlxntupde2y0o5kb3v
RIPEMD128:	f4930ef9ebfe447e257eb2d198208036
RIPEMD160:	ddfe9964a5f8d6f74d35dbde9b3a54f1735bc8b8
RIPEMD256:	b41246e0b37bf0850cbdb25ff0533f47521bd881400dd37324b3d5ebdb1c9ab1
RIPEMD320:	7c0e62706463cf42fb9ec39f6704dbb394f3ab923b6bce1668c77c735abaaf7f0c52e37eb1b99a6e

SALSA10: a8479478dff32a38838d5247df061f2f7c27e3e34864c1c4770de493ff0304bd10e3589b08
a9efb34f91747cb7b0d40b57cb9d914a463a543a942e2a53c5f5d6
SALSA20: cdf1003bca1ac07ddafa7852046d388dec243db692bc886b2af885dc4d24af0d0480d8d88817fbf3000
f429fe9ff533334e50756af762eeb61d7215d0cd428e0
SHA1: 6E017B5464F820A6C1BB5E9F6D711A667A80D8EA
SHA224: eaac129b3e54747022f5c0f15992208d8f2dc5c0e6f07b371877c7a9
SHA256: 56b1db8133d9eb398aabd376f07bf8ab5fc584ea0b8bd6a1770200cb613ca005
SHA384: 6398955d1269f368895d38ab071b623cc9ac1c024b54a8cfe79ee2250ae90551867879cac3d5e4d535f6
8ca15f134eb5
SHA512: 021b5b2ac76d969722e8b55d88d7a0c83dc61a73fd15abc98a3538a5154d7e8f0e367d771810cba13
341f7ade9818e477dd557b233112a3b14fae374f0d25098
SNEFRU: e17e71449a85dc1bcf57f8625fb1bc2ec356db707d35d630459cbb97e1f16a4e
SNEFRU256: e17e71449a85dc1bcf57f8625fb1bc2ec356db707d35d630459cbb97e1f16a4e
WHIRLPOOL: 923241d7f21e9c013d17700d820083da5daf2d123a0d256d595ea5503e8487d446644989047c9317810f7
eba097cbe50c716f91e251d92d27ae15a8cb5b7f991
HAVAL128,3: 8c6ff0690aa2b341ed5deef871bc6ed9
HAVAL160,3: 67d2bca8c7666f9e47e617e4ea8ef59becf7f282
HAVAL192,3: 0a0b06caf5648f717e584ae3a1f87188be2284e4ee44a348
HAVAL224,3: e0dcb2452bc9ebcd100d790a2bf11183cb0de8fd87b2d85b8ad7466f
HAVAL256,3: ef93bf1e9ac201453ad4b2372637ad86fac3798de8d864f0b503e8ca12b0cff4
HAVAL128,4: 0af307fac3a3b7d7dd171583ded92b1f
HAVAL160,4: 075d79f1cbcbda53b56c174571c0e86217d2f55e
HAVAL192,4: 10b25f070ef24245a0c67956cc19079b90b1b3b39a197762
HAVAL224,4: ed1e05ff9dd76bc30cecc98e49674329bbae399d30c2f30e289fd892
HAVAL256,4: 5a9f4d9415f3739cff7f5563307d59f48a6c8400b8a489a67378a0434cfaf462
HAVAL128,5: 323707af99c53d2308f64f8a595dd395
HAVAL160,5: f9138eb0c0c8f78ce16dc30bf5a4cf4d287251cd
HAVAL192,5: 0aeb3b0fb812d5085f93278ba3853a276945920599b698a7
HAVAL224,5: 4e0b9c6fd0828641245db33e99e88974a794f39017a605cb72b6949d
HAVAL256,5: 5b0e1fed54a3eacb309c7fbfb0438a952848fc88eac4d69d46e304979bc7fe51
TIGER128,3: 95434171dfe5929b3a13e0d5781bf166
TIGER160,3: 95434171dfe5929b3a13e0d5781bf166c6e364c7
TIGER192,3: 95434171dfe5929b3a13e0d5781bf166c6e364c7dd088816
TIGER128,4: a73452033e1e1733ed6a487c5f2243a1
TIGER160,4: a73452033e1e1733ed6a487c5f2243a1547c009e
TIGER192,4: a73452033e1e1733ed6a487c5f2243a1547c009ec6a1615d

Jak se ubránit prolomení hashe

Nyní by již každému mělo být jasné, že pouhé zahashování hesel uložených v databázi, prostě nestačí. Pro to, aby vaše heslo zůstalo utajeno i v případě kompromitace databáze, je ovšem možné něco udělat. A to jak na straně samotného uživatele, tak na straně provozovatele aplikace. Pojďme se proto nyní podívat na některá doporučení, která by měly obě strany dodržovat.

Co může udělat provozovatel aplikace

Solení

Zkusme si položit otázku, co se stane, když více uživatelů použije v aplikaci stejné heslo. V takovém případě budou mít všichni tito uživatelé také stejný otisk jejich hesla a útočníkovi pak stačí, aby prolomil pouze jeden hash a získá tím hesla více uživatelů současně. Sůl neboli salt, by měl být jedinečný řetězec přidružený k uživateli, o který se ještě před zahashováním heslo rozšíří. I když pak použije více uživatelů stejné heslo, budou jejich otisky díky různé hodnotě soli odlišné. Sůl navíc rozšiřuje heslo o svou délku a prolomení hesla způsoby, které jsme si uvedli, tím značně komplikuje.

V praxi můžeme heslo osolit například jménem uživatele, nebo datem jeho registrace. Je důležité pouze to, aby byla hodnota soli dostupná aplikaci ve chvíli, kdy autentizuje uživatele. V případě uživatele *Petr Novák*, který má heslo *mypass*, by po osolení vznikal uložený hash

například z řetězce *mypass_Petr_Novák*, a to jak sami uznáte, bude na prolomení mnohem obtížnější než samotné heslo *mypass*.

Pro další zvýšení bezpečnosti by se dalo doporučit ještě následné solení tajným řetězcem, který je sice znám aplikaci, ale není uložen v databázi. Aplikace tento řetězec vyčte například z konfiguračního souboru, nebo je napevno uveden jako parametr hashovací funkce.

Multi hashování (key stretching)

Další možností, jak může tvůrce aplikace zkomplikovat útočníkům případné prolamování hashí, je zahashovat heslo několikanásobně, přičemž při opakovaném hashování nemusí být použito vždy stejného algoritmu. Heslo by se mohlo zahashovat například touto sekvencí hashovacích funkcí: *md5(sha1(md5($password)))*
Účinnou metodou je také hashování v cyklu, kdy je počet aplikací hashovacího algoritmu u různých uživatelů různý. Tvůrce aplikace může použít buď náhodný počet průchodů, s tím, že si tento počet následně uloží do databáze společně s hashem, nebo může být počet průchodů závislí například na počtu znaků hesla.

Při tomto způsobu hashování odřízneme útočníka od většiny dostupných nástrojů na crackování hashů. Na své si nepřijdou dokonce ani tolik obávané rainbow tabulky, protože i ty by musely být vygenerovány stejným postupem. Útočníkovi tedy v tomto případě nezbývá než vytvořit si vlastní crackovací nástroj, který použije stejného algoritmu a bude hesla zkoušet prolomit slovníkovým útokem nebo hrubou silou. Vzhledem k tomu, že každá z hashovacích funkcí spotřebuje pro své výpočty nějaký ten čas, několikanásobně tím také prodloužíme čas, který útočník pro výpočet a porovnání hashe potřebuje a to nemluvím o tom, že útočník musí znát sekvenci funkcí, které jsou za výsledný hash zodpovědné.

Použitý hashovací algoritmus

I když byl ještě nedávno (a s největší pravděpodobností stále ještě je) nejpoužívanějším hashovacím algoritmem MD5, odborníci již řadu let upozorňují na jeho nedostatečnou odolnost. Místo něj by vývojáři měli raději používat silnější algoritmy, jako SHA-2 nebo od roku 2012 zdokonalený algoritmus SHA-3, který pro svůj výpočet potřebuje mnohem více strojového času.

Nepřenášet plaintext hesla po síti

I když bude heslo v databázi uloženo sebebezpečněji, stále zde hrozí riziko, že bude odchyceno na síti během jeho přenosu od uživatele na server v době přihlašování. Tato skutečnost sice nespadá přímo do problematiky, které je tento článek určen, ale přesto stojí za zmínku. Tvůrci aplikací by proto měli použít nějaký šifrovací algoritmus, který heslo prožene alespoň jednou hashovací funkcí, už na straně klienta. Tím se zajistí, že ani při odposlechu nedojde k vyzrazení hesla.

Omezení rychlosti a blokování uživatelů

Ještě na chvíli odhlédneme od předpokladu, že došlo k úniku dat z databáze, a že se útočník pokouší prolomit hashe na svém vlastním systému. I v případě, kdy k žádnému úniku dat nedošlo, se totiž útočník může pokusit o získání Vašeho hesla tím, že bude praktiky Brutte force nebo Dictionary attacku aplikovat přímo na přihlašovací formulář aplikace. Každá aplikace by proto měla s touto možností útoku počítat, a měla by se jí umět účinně bránit.
Jednou z možností, kterou aplikace pro svou ochranu může použít, je drobné několikamilisekundové zpoždění ve své reakci na přihlášení. Uživatel, který se legitimně přihlašuje, tuto prodlevu vůbec nepostřehne, pro automatizované nástroje, které se snaží heslo

prolomit, to ale představuje překážku, která několikanásobně zpomalí celý útok. Opatří-li vývojář aplikaci současně několikaminutovým zablokováním uživatele po několika neplatných pokusech o přihlášení, pak se uvedené útoky stávají téměř bezzubé.

Vývojáři by si v této oblasti ovšem měli počínat velice opatrně a měli by si uvědomovat možné následky v podobě DoS (odepření služeb). Pokud by například po neúspěšných pokusech o přihlášení zablokovali účet uživatele, ke kterému je zkoušeno přihlášení, pak by se ke svému účtu nemohl přihlásit ani jeho legitimní uživatel. Blokace by tedy měla probíhat na IP adresu, ze které neúspěšné požadavky na přihlášení přicházejí a měla by být pouze ve spojitosti s konkrétním účtem, aby se ke svým účtům mohli přihlásit ostatní uživatelé ze stejné vnitřní sítě.

Co může udělat koncový uživatel

Síla hesla

Jeden z aspektů, který hraje velkou roli při zkomplikování (znemožnění) získání hesla k odpovídajícímu hashi je jeho kvalita neboli síla. Z odstavců o prolomení metodami brutte force, dictionary attack a s použitím rainbow tabulek vyplývá, že chceme-li útočníkovi zabránit v odhalení hesla po úniku hashů, musíme své heslo zvolit tak, aby se neskládalo jen z prostých slov, která je možné najít v některém slovníku. Nepomůže ani přidání číselných prefixů nebo postfixů. Dále je důležité, aby heslo obsahovalo znaky z různých množin, tzn. čísla, malá písmena, velká písmena a speciální znaky. V neposlední řadě pak o úspěchu rozhoduje délka hesla. V dnešní době bych doporučil používat hesla dlouhá alespoň 12 znaků. A jak by tedy mělo takovéto silné heslo nakonec vypadat?

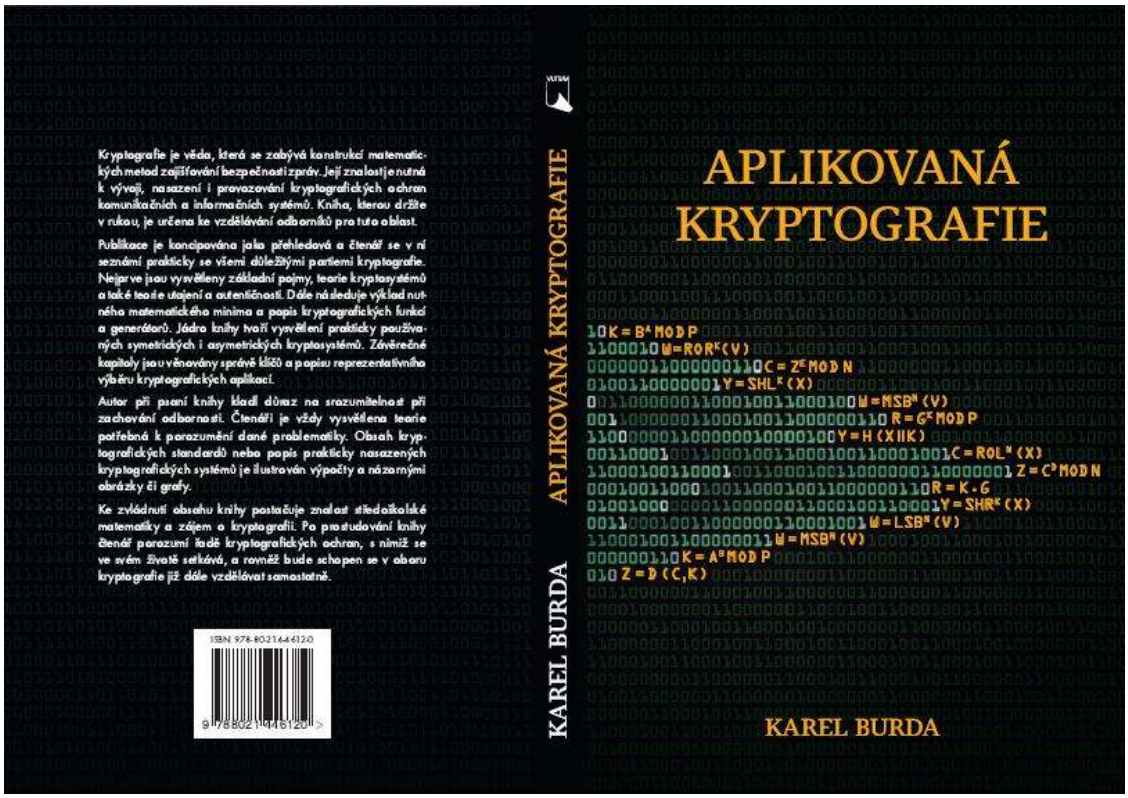
Například: *S2ppo,pzl.Szn1m,pnk.*

Nejspíš si řeknete, kdo si má takové heslo zapamatovat, když psaní hesel na lístečky a jejich lepení na monitor, není rovněž ideálním řešením. Stačí si však udělat jednoduchý systém pro tvorbu hesel a uvidíte, že to zase taková věda není. Například uvedené heslo *S2ppo,pzl.Szn1m,pnk.* jsou první písmena slov z mírně upravené říkanky *„Skákali dva psi přes pole, přes zelenou louku. Šel za nimi jeden myslivec, péro na klobouku.“*

Nepoužívat stejné heslo na různých místech

Aby toho ale nebylo málo, zapamatovat si jedno univerzální heslo v žádném případě nestačí. Pokud byste totiž používali stejné heslo u více služeb a jedna z nich by ukládala v databázi hesla ve formě prostého textu, pak by v případě úniku dat získal útočník přístup i na vaše účty v těch aplikacích, které se o bezpečné uložení dokáží postarat. O tom jsem se ale zmiňoval již na samém začátku tohoto textu. Používání více různých hesel není tedy v žádném případě jen jakousi buzerací uživatelských mozků, ale opodstatněná záležitost, kterou byste měli bezpodmínečně dodržovat.

C. Upoutávka na knihu K.Burdy – Aplikovaná kryptografie



Nakladatelství VUTIUM vydalo toto září knihu docenta vysokoho technického učení v Brně doc. Ing. Karel Burdy, CSc.: Aplikovaná kryptografie.

Obdobná česká kniha není v současné době na trhu dostupná, a tak se domnívám, že by tato informace mohla čtenáře našeho e-zinu Vašeho zajímat.
Kniha v ceně cca 399,- Kč má rozsah 250 stran. Obsahuje úvod do teoretické kryptografie a uceleně pokrývá problematiku aplikované kryptografie.
Publikace je koncipována jako přehledová a čtenář se v ní seznámí čtivou formou prakticky se všemi důležitými partiemi kryptografie. Nejprve jsou v knize vysvětleny základní pojmy, teorie kryptosystémů a také teorie utajení a autentičnosti.
Dále následuje matematické minimum potřebné k porozumění kryptografickým konstrukcím a popis kryptografických generátorů a jednosměrných funkcí.
Jádro knihy tvoří vysvětlení prakticky používaných symetrických i asymetrických kryptosystémů. Závěrečné kapitoly jsou věnovány správě klíčů a popisu reprezentativního výběru kryptografických aplikací.
Autor při psaní knihy kladl důraz na srozumitelnost při zachování odbornosti. Čtenáři je vždy nejprve vysvětlena potřebná teorie k porozumění dané problematiky. Popis kryptografických standardů nebo popis prakticky nasazených kryptografických systémů, je ilustrován výpočty a názornými obrázky či grafy.
Považuji za cenné, že ke zvládnutí obsahu této knihy postačuje znalost středoškolské matematiky a zájem o kryptografii. Čtenář po prostudování knihy porozumí řadě kryptografických ochran, s nimiž se ve svém životě setkává a rovněž bude schopen se v oboru kryptografie již dále vzdělávat samostatně.

S laskavým souhlasem autora naleznete v příloze k tomuto e-zinu ukázky jednotlivých kapitol.

D. Soutěž v luštění / Dosud nevyluštěný dopis českého pobělohorského emigranta Karla Rabenhaupta ze Suché z doby třicetileté války.

Jakub Mírka, SOA Plzeň, mirka@soaplzen.cz
Pavel Vondruška, pavel.vondruska@crypto-world.info

Vážení soutěžící, děkujeme za váš zájem a účast v letošní soutěži v luštění, kterou jsme vyhlásili v minulém čísle našeho e-zinu a pro jejíž podporu byla vytvořena doprovodná stránka http://soutez2013.crypto-world.info/ .

Bohužel se ukázalo, že předložený hlavní úkol – vyluštit dosud nevyluštěný dopis z doby třicetileté války (konkrétně dopis českého pobělohorského emigranta Karla Rabenhaupta ze Suché) je příliš těžkým oříškem a dosud nikdo ze soutěžících jej nedokázal vyřešit.

Musím také přiznat, že ani předkladatelé této úlohy prozatím řešení neznají, a tak se neobjevili ani nápovědy, které by pomohli v hledání řešení.

Vyhlašovatelé soutěže se proto dohodli, že letos bude soutěž ukončena k datu 1. 12. 2013 (pokud ovšem někdo přece jen do tohoto data neohlásí, že dokázal dopis vyluštit).

První cenu v takovém případě získá řešitel, který zašle v termínu do 3. 12. 2013 své poznámky o hledání řešení úlohy (ovšem již bez bezplatné účasti na mezinárodním kryptologickém workshopu Mikulášská kryptobesídka).

Poznámky by měly být zpracovány ve tvaru článku a popisovat aktivity řešitele, jaké výsledky získal, ověřil apod.

Ze zaslaných textů vybere „komise“ ve složení J. Mírka, P. Vondruška článek ke zveřejnění v prosincovém čísle e-zinu Crypto-Worldu 11-12/2013.

Hlavním kritériem hodnocení bude, zda se autor přiblížil k předpokládanému řešení a vedlejším kritériem čtivost a zajímavost zaslaného článku.

Vybraný autor článku se stane vítězem letošní soutěže a za odměnu získá tablet GOCLEVER TAB R76.2 včetně pouzdra s klávesnicí, kterou věnovala firma DIGNITA, s.r.o.

Články s popisem hledání řešení a získanými částečnými výsledky zasílejte na e-mail obou hodnotitelů a to do půlnoci 3. 12. 2013, předmět Rabenhaupt.
Délka článku (včetně obrázků, grafů a tabulek) nesmí být delší než 15 stran! Je však možné přiložit libovolný počet příloh.

Těšíme se na zaslané texty!

S pozdravem
Pavel Vondruška
Jakub Mírka

E. O čem jsme psali za posledních 12 měsíců

Kompletní obsah všech vyšlých čísel od roku 1999 je dostupný zde
http://crypto-world.info/index2.php?vyber=obsah

Crypto-World 5-6/2012

A.	HERMANN POKORNY - "zaslúžilý umelec" v lúštiteľskom odbore vo víre I. svetovej vojny (J.Krajčovič)	2 - 8
B.	Najstaršia zašifrovaná písomná pamiatka v Čechách (J.Krajčovič)	9 – 10
C.	Nízkoriziková kryptografie (V.Klíma)	11 - 13
D.	Společná novela zákona o elektronickém podpisu (účinná od 1.7.2012) (P.Vondruška)	14 – 18
E.	Call for Papers - Mikulášská kryptobesídka 2012	19
F.	O čem jsme psali v květnu a v červnu 2000 – 2011	20 – 24
G.	Závěrečné informace	25

Crypto-World 7-8/2012

A.	Andreas Figl – Nestor rakúskej školy kryptológie	2 – 13
B.	Kryptologické perličky 1 (K.Šklíba)	14 – 24
C.	Z NISTu unikl interní dokument k SHA-3 (V.Klíma)	25 - 30
D.	Kniha Kryptologie, šifrování a tajná písma rozebrána (P.Vondruška)	31
E.	Problematika infrastruktury veřejných klíčů (PKI), kurz Akademie CZ.NIC (P.Vondruška)	32 – 23
F.	ZPRÁVA - Nechcete být odposlouchávání? (L.Stejskalová)	34
G.	O čem jsme psali v létě 2000 – 2011	35 – 37
H.	Závěrečné informace	38

Příloha: dokument, který měl být odeslán pouze do interní skupiny NISTu pro výběr SHA-3 (více informací viz článek V.Klímy)

Crypto-World 9-10/2012

A.	Pointerová šifra a nízkoriziková náhrada AES (V.Klíma)	2 – 8
B.	Kryptografické zabezpečení prodeje lihovin (R.Palovský)	9 – 13
C.	Kryptologické perličky 2 (K.Šklíba)	14 – 20
D.	Záhada kodexu Rohonczi Codex (E. Antal)	21 – 28
E.	Kaspersky Lab uvádí Kaspersky Internet Security 2013	29 - 31
F.	O čem jsme psali v září a říjnu 1999 – 2011	32 – 35
G.	Závěrečné informace	36

Příloha: neoglyfy.pdf , ukázka ze sešitu Batěk A. S.: Neoglyfy I., str. 4 – str. 13 (http://crypto-world.info/casop14/neoglyfy.pdf)

Crypto-World 11-12/2012

A.	SHA-3 a lehká kryptografie (V.Klíma)	2 – 11
B.	Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni , část I. (J.Mírka)	12 – 28
C.	Tip na vánoční dárek - Enigma - bitva o kód (P.Vondruška)	29 – 30
D.	Pracovní příležitost (World Startup Project)	31
E.	O čem jsme psali v listopadu a prosinci 1999 – 2011	32 – 35
F.	Závěrečné informace	36

Příloha: Obrazová příloha k článku B (Mírka, J.) http://crypto-world.info/casop14/cast1.zip

Crypto-World 1-2/2013

A. Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni, část II. (J.Mírka) 2 -12
B. Lúštitelia historických šifier - A.V. Maloch a Josef Šusta (J. Krajčovič) 13 - 21
C. Elektronický podpis v praxi (P.Vondruška, J.Peterka) 22
D. SOOM.cz - Hacking & Security konference #2 (R.Kümmel) 23
E. Security and Protection of Information 2013 24 – 25
F. O čem jsme psali za posledních 12 měsíců 26 - 27
G. Závěrečné informace 28

Příloha: Obrazová příloha k části II. Mírka, J.: Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni
http://crypto-world.info/casop15/obr2.zip

Crypto-World 3-4/2013

A. Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni, část III. (J.Mírka) 2 -14
B. Andreas Figl – rakúsky dôstojník a kryptológ (J.Kollár) 15 - 23
C. Central European Conference on Cryptology 2013 24
D. call for papers - CYBERSPACE 2013 25 - 26
E. O čem jsme psali za posledních 12 měsíců 27 - 28
F. Závěrečné informace 29

Příloha: Obrazová příloha k části III. Mírka, J.: Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni
http://crypto-world.info/casop15/obr3.zip

Crypto-World 5-6/2013

A. Konec aktualit v Crypto-News a Bezpečnostních střípků (J.Pinkava) 2
B. Tajomstvo šifrovacieho stroja G. W. Leibniza (J.Krajčovič) 3 – 11
C. Kaspersky Lab odhalila novou kyberšpionážní operaci NetTraveler 12
D. Reakcia na článok „Andreas Figl – rakúsky dôstojník a kryptológ“ (J.Krajčovič) 13 – 15
E. Cvičný CISSP test z kryptografie 16 – 18
F. Central European Conference on Cryptology 2013 26.-28. června, Telč 19 – 20
G. Call for Papers Mikulášská kryptobesídka 21
H. O čem jsme psali za posledních 12 měsíců 22
I. Závěrečné informace 23

Crypto-World 7-8/2013

A. Reino Häyhänen – sovietsky špión (J. Kollár) 2 – 9
B. Dosud nevyluštěný dopis českého pobělohorského emigranta Karla Rabenhaupta ze Suché z doby třicetileté války. (Soutěž o ceny) (J. Mírka) 10 – 18
C. Soutěž 2013, luštění originálního šifrového dopisu ze 17. století (P.Vondruška)19 – 21
D. Diskrétní logaritmus a metody jeho výpočtu (J. Pulec) 22 – 26
E. Kaspersky v Praze - Kybernetické zbraně jsou nejhorším vynálezem století 27 – 28
F. Pozvánka k podzimním kurzům Akademie CZ NIC 29 – 31
G. O čem jsme psali za posledních 12 měsíců 32 – 33
H. Závěrečné informace 34

F. Závěrečné informace

1. Sešit

Crypto-World je oficiální informační sešit "***Kryptologické sekce Jednoty českých matematiků a fyziků***" (GCUCMP). Obsahuje články podepsané autory. Případné chyby a nepřesnosti jsou dílem autorů jednotlivých podepsaných článků, GCUCMP za ně nemá odbornou ani jinou zodpovědnost.
Adresa URL, na níž můžete najít tento sešit (zpravidla 3 týdny po jeho rozeslání) a předchozí e-ziny, denně aktualizované novinky z kryptologie a informační bezpečnosti, normy, standardy, stránky některých členů a další související materiály: http://crypto-world.info

2. Registrace / zrušení registrace

Zájemci o e-zin se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@crypto-world.info (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci elektronický formulář na http://crypto-world.info. Při registraci vyžadujeme pouze jméno a příjmení, titul, pracoviště (není podmínkou) a e-mail adresu určenou k zasílání kódů ke stažení sešitu.
Ke zrušení registrace stačí zaslat krátkou zprávu na e-mail pavel.vondruska@crypto-world.info (předmět: ruším odběr Crypto-Worldu!) nebo opět použít formulář na http://crypto-world.info. Ve zprávě prosím uveďte jméno a příjmení a e-mail adresu, na kterou byly kódy zasílány.

3. Redakce

E-zin Crypto-World

Redakční práce:	Pavel Vondruška
	Jozef Krajčovič
	Jozef Martin Kollar
	Vlastimil Klíma
Přehled autorů:	http://crypto-world.info/obsah/autori.pdf
Webmaster	Pavel Vondruška, jr.

4. Spojení (abecedně)

redakce e-zinu	ezin@crypto-world.info ,	http://crypto-world.info
Vlastimil Klíma	v.klima@volny.cz ,	http://cryptography.hyperlink.cz/
Jozef Martin Kollar	jmkollar@math.sk ,	
Jozef Krajčovič	kryptosvet@gmail.com ,	http://katkryptolog.blogspot.sk
Jaroslav Pinkava	jaroslav.pinkava@gmail.com ,	http://crypto-world.info/pinkava/
Pavel Vondruška	pavel.vondruska@crypto-world.info	http://crypto-world.info/vondruska/index.php
Pavel Vondruška, jr.	pavel@crypto-world.info ,	http://www.pavelvondruska.cz/